

# **PREDICTION OF ELECTRICITY BILL USING SML TECHNIQUE**

Submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering  
Technology degree in Information Technology

By

**VIGNESH G (38120090)**

**ROSHAN CHRISTY R (38120069)**



**DEPARTMENT OF INFORMATION TECHNOLOGY  
SCHOOL OF COMPUTING**

# **SATHYABAMA**

**INSTITUTE OF SCIENCE AND TECHNOLOGY  
(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade "A" by NAAC**

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119**

**MAY - 2022**



**SATHYABAMA**

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

[www.sathyabama.ac.in](http://www.sathyabama.ac.in)

## DEPARTMENT OF INFORMATION TECHNOLOGY

### **BONAFIDE CERTIFICATE**

This is to certify that this Project Report is the bonafide work of **VIGNESH G (38120090), ROSHAN CHRISTY R (38120069)** who carried out the project entitled “ **PREDICTION OF ELECTRICITY BILL USING SML TECHNIQUE** ” under our supervision from

**Dr. P. JEYANTHI M.E.,PH.D.,**

**Internal Guide**

**Dr. R.**

**SUBHASHINI**

**M.E., Ph.D.,**

**Head of the**

**Department**

Submitted for Viva voce Examination held on \_\_\_\_\_

**Internal Examiner**

**External Examiner**

**DECLARATION**

I **VIGNESH (38120090)** hereby declare that the Project Report entitled “**PREDICTION OF ELECTRICITY BILL USING SML TECHNIQUE**” done by me under the guidance of **Dr. P. JEYANTHI M.E., Ph.D.**, is submitted in partial fulfilment of the requirements for the award of Bachelor of Technology degree in Information Technology.

**DATE:**

**PLACE:**

**SIGNATURE OF THE CANDIDATE**

## **ACKNOWLEDGEMENT**

**I am pleased to acknowledge my sincere thanks to Board of Management of SATHYABAMA for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.**

**I convey my thanks to Dr. T. Sasikala M.E., Ph.D., Dean, School of Computing and Dr.R.Subhashini M.E.,Ph.D., Head of the Department of Information Technology for providing me necessary support and details at the right time during the progressive reviews.**

**I would like to express my sincere and deep sense of gratitude to my Project Guide Dr. P. JEYANTHI M.E., Ph.D., for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.**

**I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of Information Technology who were helpful in many ways for the completion of the project.**

## ABSTRACT

Generally, predicting how the Electricity price will perform is one of the most difficult things to do. It can be described as one of the most critical process to predict that. This is a very complex task and has uncertainties. To prevent this problem in One of the most interesting (or perhaps most profitable) time series data using machine learning techniques. Hence, Electricity price prediction has become an important research area. The aim is to predict machine learning based techniques for Electricity price prediction results in best accuracy. The analysis of dataset by supervised machine learning technique(SMLT) to capture several information's like, variable identification, uni-variate analysis, bi-variate and multi-variate analysis, missing value treatments and analyze the data validation, data cleaning/preparing and data visualization will be done on the entire given dataset. To propose a machine learning-based method to accurately predict the Electricity price Index value by prediction results in the form of Electricity price increase or stable state best accuracy from comparing supervise classification machine learning algorithms. Additionally, to compare and discuss the performance of various machine learning algorithms. dataset with evaluation classification report, identify the confusion matrix and to categorizing data from priority and the result shows that the effectiveness of the proposed machine learning algorithm technique can be compared with best accuracy MAE,MSE, $R^2$ .

## TABLE OF CONTENT

| CHAPTER<br>NO. | TITLE                               | PAGE NO.  |
|----------------|-------------------------------------|-----------|
|                | ABSTRACT                            | i         |
|                | LIST OF FIGURES                     | iv        |
|                | LIST OF TABLES                      | v         |
| <b>1.</b>      | <b>INTRODUCTION</b>                 | <b>1</b>  |
|                | 1.1 EXISTING SYSTEM                 | 2         |
|                | 1.2 DISADVANTAGES                   | 2         |
| <b>2.</b>      | <b>LITERATURE SURVEY</b>            | <b>3</b>  |
| <b>3.</b>      | <b>AIM AND SCOPE OF THE PROJECT</b> | <b>6</b>  |
|                | 3.1 SYSTEM STUDY                    | 6         |
|                | 3.1.1 OBJECTIVES                    | 6         |
|                | 3.1.2 PROJECT GOALS                 | 6         |
|                | 3.1.3 SCOPE OF THE PROJECT          | 7         |
|                | 3.2 FEASIBILITY STUDY               | 7         |
|                | 3.3 PROJECT REQUIREMENTS            | 8         |
|                | 3.4 WORKING PROCESS                 | 9         |
| <b>4.</b>      | <b>METHODOLOGIES</b>                | <b>11</b> |
|                | 4.1 SOFTWARE DESCRIPTION            | 11        |
|                | 4.1.1 MACHINE LEARNING              | 11        |
|                | 4.1.2 ANACONDA NAVIGATOR            | 12        |
|                | 4.1.3 JUPITER NOTEBOOK              | 13        |
|                | 4.1.4 PYTHON                        | 13        |
|                | 4.2 SYSTEM ARCHITECTURE             | 14        |
|                | 4.3 MODULE DESCRIPTION              | 16        |
|                | 4.3.1 DATA PRE-PROCESSING           | 16        |

|           |                                                                             |           |
|-----------|-----------------------------------------------------------------------------|-----------|
| 4.3.2     | DATA VALIDATION/CLEANING/PREPARING PROCESS                                  | 18        |
| 4.3.3     | EXPLORATION DATA ANALYSIS OF VISUALIZATION                                  | 19        |
| 4.3.4     | COMPARING ALGORITHM WITH PROTECTION IN THE FORM<br>OF BEST ACCURACY RESULTS | 21        |
| 4.3.5     | REGRESSION                                                                  | 23        |
| 4.4       | ALGORITHM AND TECHNIQUES                                                    | 25        |
| 4.4.1     | SUPPORT VECTOR REGRESSION                                                   | 26        |
| 4.4.2     | DECISION TREE REGRESSION                                                    | 27        |
| 4.4.3     | RANDOM FOREST REGRESSION                                                    | 29        |
| 4.4.4     | LINEAR REGRESSION                                                           | 30        |
| 4.4.5     | LASSO REGRESSION                                                            | 32        |
| 4.4.6     | RIDGE REGRESSION                                                            | 33        |
| 4.5       | DEPLOYMENT                                                                  | 35        |
| 4.5.1     | FLASK (WEB FRAMEWORK)                                                       | 35        |
| 4.5.2     | HTML                                                                        | 36        |
| 4.5.3     | CSS                                                                         | 36        |
| <b>5.</b> | <b>RESULTS AND DISCUSSION</b>                                               | <b>37</b> |
| 5.1       | RESULT                                                                      | 37        |
| 5.2       | DISCUSSION                                                                  | 38        |
| <b>6.</b> | <b>CONCLUSION AND FUTURE WORK</b>                                           | <b>39</b> |
| 6.1       | CONCLUSION                                                                  | 39        |
| 6.2       | FUTURE WORK                                                                 | 39        |
| <b>7.</b> | <b>REFERENCES</b>                                                           | <b>40</b> |

## LIST OF FIGURES

| FIGURE NO. | TITLE                              | PAGE NO. |
|------------|------------------------------------|----------|
| 3.1        | PROCESS OF DATA FLOW DIAGRAM       | 8        |
| 4.1        | PROCESS OF MACHINE LEARNING        | 11       |
| 4.2        | ANACONDA NAVIGATOR                 | 12       |
| 4.3        | SYSTEM ARCHITECTURE                | 14       |
| 4.4        | WORKFLOW DIAGRAM                   | 15       |
| 4.5        | USE CASE DIAGRAM                   | 16       |
| 4.6        | MODULE 1 DATA VALIDATION           | 19       |
| 4.7        | MODULE 1 DIAGRAM                   | 19       |
| 4.8        | MODULE 2 VISUALIZATION             | 20       |
| 4.9        | MODULE 2 DIAGRAM                   | 21       |
| 4.10       | MODULE 3 SUPPORT VECTOR REGRESSION | 27       |
| 4.11       | MODULE 3 DECISION TREE REGRESSOR   | 28       |
| 4.12       | MODULE 3 DIAGRAM                   | 28       |
| 4.13       | MODULE 4 RANDOM FOREST REGRESSOR   | 30       |
| 4.14       | MODULE 4 LINEAR REGRESSION         | 31       |
| 4.15       | MODULE 4 DIAGRAM                   | 32       |
| 4.16       | MODULE 5 LASSO REGRESSION          | 33       |
| 4.17       | MODULE 5 RIDGE REGRESSION          | 34       |
| 4.18       | MODULE 5 DIAGRAM                   | 34       |
| 5.1        | OUTPUT SCREENSHOT                  | 37       |
| 5.2        | OUTPUT SCREENSHOT                  | 38       |

## LIST OF TABLES

| TABLE NO.<br>NO. | TITLE           | PAGE |
|------------------|-----------------|------|
| 4.1              | ATTRIBUTES      | 18   |
| 5.1              | ACCURACY RESULT | 37   |

## **CHAPTER-1 INTRODUCTION**

The power market has been drawing in many individuals and its significance. The start of the power value forecast can be viewed as in the mid-'90s. It depends entirely on the nature and nature of human exercises. One-of-a-kind highlights and makes helpful costs noticeable in different business sectors. Know the idea of the day, the week, the times, the amazements, the brief time frame, and frequently the costs are not relied upon to rise. Furthermore, the potential circuits for distinguishing this item are not set in stone similarly and are hard to pinpoint. Power costs (EPF) indicate land-based energy and future expenses for some requirements, which has added to independent direction and interest in energy organizations. In this manner, various EPF-related issues should be tended to, generally regarding financial aspects. Conventional writing presents three sorts of short-, medium-, and long-haul thoughts, which are distinct by the craft of tackling monetary issues. Regularly, it very well may be viewed as transitory: from a couple of moments to a couple of days prior to the proceeding; medium: the period from a couple of days to a while; life span: a time of a couple of months to numerous years.

By and large, the weather conditions are the initial two sorts that accentuate the day's predictions. Notwithstanding, as referenced above, it is vital to comprehend that one level is significant; indeed, the film is set to repeat its part in another episode. As indicated by a unique interest plan, it can utilize an assortment of strategies for show. Specifically, most of the techniques utilized fall into the accompanying five classifications: factual construction, astute figuring strategies, character decrease, needs, and different models. This investigation centers around the initial two families and sums up them with a blend of thoughts that feature the numerous advancements that have been composed. To keep up, how about we initially set up conceivable thoughts. Booking articles are expected to show the energy utilized at a specific season of the day, ordinarily temporarily. The high level is significant toward the end. As of late, the past work has lost its importance to the alleged theory speculation, which depends on distance forecast to distinguish 38 amounts, as well as expectation

of its predominance.

## **1.1 EXISTING SYSTEM:**

Electricity market participants rely on data-driven methods using public market data to predict locational marginal prices (LMPs) and determine optimal bidding strategies, since they cannot access confidential power system models and operating details. In this paper, system-wide heterogeneous public market data are organized into a 3-dimensional (3D) tensor,  $N \times N \times N$  which can store their spatio-temporal correlations. To predict real-time locational marginal prices (RTLMPs) by learning the spatio-temporal correlations stored in the historical market data tensor. An autoregressive moving average (ARMA) calibration method is adopted to improve the prediction accuracy. Case studies using public market data from Midcontinent Independent System Operator (MISO) and Southwest Power Pool (SPP) demonstrate that the proposed method is able to learn spatio-temporal correlations among RTLMPs and perform accurate RTLMP prediction. Another potential area of exploration includes predicting energy price component of LMP by learning total supply-demand balance. Congestion price component and loss price component may be predicted by other spatial learning methods. The final spatio-temporal correlated nodal LMPs can be assembled based on these three individual price components.

## **1.2 DISADVANTAGES:**

- They are not classifying Electricity price on machine learning classification technique and not mention any accuracy results.
- It can't thereby better determine the regularity of Electricity price prediction data and achieve more accurate prediction results.

## **CHAPTER-2**

### **LITERATURE SURVEY**

**Title :** Electricity price forecasting in deregulated markets: A review and evaluation

**Author:** Sanjeev Kumar Aggarwal, Lalit Mohan Saini, Ashwani Kumar

**Year : 2008**

The main methodologies used in electricity price forecasting have been reviewed in this paper. The following price-forecasting techniques have been covered: (i) stochastic time series, (ii) causal models, and (iii) artificial intelligence based models. The quantitative analysis of the work done by various authors has been presented based on (a) time horizon for prediction, (b) input variables, (c) output variables, (d) results, (e) data points used for analysis, (f) preprocessing technique employed, and (g) architecture of the model. The results have been presented in the form of tables for ease of comparison. Classification of various price-influencing factors used by different researchers has been done and put for reference. Application of various models as applied to different electricity markets is also presented for consideration

**Title :** An Integrated Machine Learning Model for Day-Ahead Electricity Price Forecasting

**Author:** Shu Fan, James R. Liao, Kazuhiro Kaneko, Member, IEEE, and Luonan Chen, Senior Member, IEEE

**Year : 2006**

This paper proposes a novel model for short-term electricity price forecasting based on an integration of two machine learning technologies: Bayesian Clustering by Dynamics (BCD) and Support Vector Machine (SVM). The proposed forecasting system adopts an integrated architecture. Firstly, a BCD classifier is applied to cluster the input data set into several subsets in an unsupervised manner. Then, groups of 24 SVMs for the next day's electricity price profile are used to fit the training data of each subset in a supervised way. To demonstrate the effectiveness, the proposed model

has been trained and tested on the data of the historical energy prices from the New England electricity market.

**Title** : A Review For Electricity Price Forecasting Techniques In Electricity Markets

**Author:** Ankur Jain, Ankit Tuli, Misha Kakkar

**Year** : 2013

Electricity Price forecasting has been an important and crucial issue in every nation nowadays. This paper summarizes the various proposed models and techniques for the price forecasting. Accuracy of the price forecasting significantly influence the profits for transmission companies, distributors, suppliers etc. It presents an extensive review of different approaches for price forecasting. Different approaches and methods like ARIMA (Auto Regressive Integrated Moving Average), LSSVM (Least Square Support Vector Machine), LLWNN (Local Linear Wavelet Neural Network), ANN (Artificial Neural Network) etc. are compared and weakness and strengths of these methods are analyzed. Since both the consumers and producers are dependent on price forecasting information for their bidding strategies, an efficient method is required for the same. When producer got accurate forecast, he can propose his strategy for maximization of profit. Similarly a consumer can schedule to minimize the cost of electricity.

**Title** : Price Forecasting for the Balancing Energy Market Using Machine-Learning Regression

**Author:** Alexandre Lucas, Konstantinos Pegios, Evangelos Kotsakis and Dan Clarke

**Year** : 2020

The importance of price forecasting has gained attention over the last few years, with the growth of aggregators and the general opening of the European electricity markets. Market participants manage a tradeoff between, bidding in a lower price market (day-ahead), but with typically higher volume, or aiming for a lower volume market but with potentially higher returns (balance energy market). Companies try to forecast the extremes of revenues or prices, in order to manage risk and opportunity, assigning their assets in an optimal way. It is thought that in general,

electricity markets have quasi-deterministic principles, rather than being based on speculation, hence the desire to forecast the price based on variables that can describe the outcome of the market. Many studies address this problem from a statistical approach or by performing multiple-variable regressions, but they very often focus only on the time series analysis. In 2019, the Loss of Load Probability (LOLP) was made available in the UK for the first time. Taking this opportunity, this study focusses on five LOLP variables (with different time-ahead estimations) and other quasi-deterministic variables, to explain the price behavior of a multi-variable regression model. These include base production, system load, solar and wind generation, seasonality, day-ahead price and imbalance volume contributions. Three machine-learning algorithms were applied to test for performance, Gradient Boosting (GB), Random Forest (RF) and XGBoost. XGBoost presented higher performance and so it was chosen for the implementation of the real time forecast step. The model returns a Mean Absolute Error (MAE) of 7.89 £/MWh, a coefficient of determination (R2 score) of 76.8% and a Mean Squared Error (MSE) of 124.74. The variables that contribute the most to the model are the Net Imbalance Volume, the LOLP (aggregated), the month and the De-rated margins (aggregated) with 28.6%, 27.5%, 14.0%, and 8.9% of weight on feature importance respectively.

**Title** : Energy Markets Forecasting. From Inferential Statistics to Machine Learning: The German Case

**Author:** Emma Viviani, Luca Di Persio and Matthias Ehrhardt

**Year** : 2021

In this work, we investigate a probabilistic method for electricity price forecasting, which overcomes traditional ones. We start considering statistical methods for point forecast, comparing their performance in terms of efficiency, accuracy, and reliability, and we then exploit Neural Networks approaches to derive a hybrid model for probabilistic type forecasting. We show that our solution reaches the highest standard both in terms of efficiency and precision by testing its output on German electricity prices data.

## **CHAPTER-3**

### **AIM AND SCOPE OF THE PROJECT**

#### **3.1 SYSTEM STUDY**

##### **3.1.1 Objectives**

The goal is to develop a machine learning model for Electricity Bill Prediction, to potentially replace the updatable supervised machine learning classification models by predicting results in the form of best accuracy by comparing supervised algorithm.

##### **3.1.2 Project Goals**

- Exploration data analysis of variable identification
  - Loading the given dataset
  - Import required libraries packages
  - Analyze the general properties
  - Find duplicate and missing values
  - Checking unique and count values
- Uni-variate data analysis
  - Rename, add data and drop the data
  - To specify data type
- Exploration data analysis of bi-variate and multi-variate
  - Plot diagram of pairplot, heatmap, bar chart and Histogram
- Method of Outlier detection with feature engineering
  - Pre-processing the given dataset
  - Splitting the test and training dataset

- Comparing the Decision tree and Logistic regression model and random forest etc.
- Comparing algorithm to predict the result
  - Based on the best accuracy

### **3.1.3 Scope Of The Project**

The main Scope is to detecting the Electricity Bill Prediction, which is a classic text regression problem with a help of machine learning algorithm. After we get the accuracy then we have to deploying the flask framework.

#### ***Preparing the Dataset :***

This dataset contains 1000 records extracted, which were classified into 10 Features.

## **3.2 FEASIBILITY STUDY:**

### ***Data Wrangling***

In this section of the report will load in the data, check for cleanliness, and then trim and clean given dataset for analysis. Make sure that the document steps carefully and justify for cleaning decisions.

### ***Data collection***

The data set collected for predicting given data is split into Training set and Test set. Generally, 7:3 ratios are applied to split the Training set and Test set. The Data Model which was created using Random Forest, logistic, Decision tree algorithms and Support vector classifier (SVC) are applied on the Training set and based on the test result accuracy, Test set prediction is done.

### ***Preprocessing***

The data which was collected might contain missing values that may lead to inconsistency. To gain better results data need to be preprocessed so as to improve the efficiency of the algorithm. The outliers have to be removed and also variable conversion need to be done.

### ***Building the Regression model***

The prediction of Electricity Bill, A Random Forest Algorithm prediction model is effective because of the following reasons: It provides better results in regression problem.

- It is strong in preprocessing outliers, irrelevant variables, and a mix of continuous, categorical and discrete variables.
- It produces out of bag estimate error which has proven to be unbiased in many tests and it is relatively easy to tune with.

### ***Construction of a Predictive Model***

Machine learning needs data gathering have lot of past data's. Data gathering have sufficient historical data and raw data. Before data pre-processing, raw data can't be used directly. It's used to preprocess then, what kind of algorithm with model. Training and testing this model working and predicting correctly with minimum errors. Tuned model involved by tuned time to time with improving the accuracy.

***Fig: 3.1 Process of dataflow diagram***

### 3.3 PROJECT REQUIREMENTS

#### **General:**

Requirements are the basic constraints that are required to develop a system. Requirements are collected while designing the system. The following are the requirements that are to be discussed.

1. Functional requirements
2. Non-Functional requirements
3. Environment requirements
  - A. Hardware requirements
  - B. software requirements

#### **Functional Requirements:**

The software requirements specification is a technical specification of requirements for the software product. It is the first step in the requirements analysis process. It lists requirements of a particular software system. The following details to follow the special libraries like sk-learn, pandas, numpy, matplotlib and seaborn.

#### **Non-Functional Requirements:**

Process of functional steps,

1. Problem define
2. Preparing data
3. Evaluating algorithms
4. Improving results
5. Prediction the result

#### **Environmental Requirements:**

##### **1. Software Requirements:**

Operating System : Windows  
Tool : Anaconda with Jupyter Notebook

##### **2. Hardware requirements:**

Processor : Pentium IV/III

|           |                 |
|-----------|-----------------|
| Hard disk | : minimum 80 GB |
| RAM       | : minimum 2 GB  |

### 3.4 WORKING PROCESS:

- ☐ Download and install anaconda and get the most useful package for machine learning in Python.
- ☐ Load a dataset and understand its structure using statistical summaries and data visualization.
- ☐ Machine learning models, pick the best and build confidence that the accuracy is reliable.

Python is a popular and powerful interpreted language. Unlike R, Python is a complete language and platform that you can use for both research and development and developing production systems. There are also a lot of modules and libraries to choose from, providing multiple ways to do each task. It can feel overwhelming.

When you are applying machine learning to your own datasets, you are working on a project. A machine learning project may not be linear, but it has a number of well-known steps:

- Define Problem.
- Prepare Data.
- Evaluate Algorithms.
- Improve Results.
- Present Results.

The best way to really come to terms with a new platform or tool is to work through a machine learning project end-to-end and cover the key steps. Namely, from loading data, summarizing data, evaluating algorithms and making some predictions.

Here is an overview of what we are going to cover:

1. Installing the Python anaconda platform.
2. Loading the dataset.
3. Summarizing the dataset.
4. Visualizing the dataset.
5. Evaluating some algorithms.
6. Making some predictions.

## **CHAPTER - 4**

### **METHODOLOGIES**

#### **4.1 SOFTWARE DESCRIPTION**

##### **4.1.1 Machine Learning**

Machine learning is to predict the future from past data. Machine learning (ML) is a type of artificial intelligence (AI) that provides computers with the ability to learn without being explicitly programmed. Machine learning focuses on the development of Computer Programs that can change when exposed to new data and the basics of Machine Learning, implementation of a simple machine learning algorithm using python. Process of training and prediction involves use of specialized algorithms. It feed the training data to an algorithm, and the algorithm uses this training data to give predictions on a new test data. Machine learning can be roughly separated in to three categories. There are supervised learning, unsupervised learning and reinforcement learning.

Supervised learning program is both given the input data and the corresponding labeling to learn data has to be labeled by a human being beforehand. Unsupervised learning is no labels. It provided to the learning algorithm. This algorithm has to figure out the clustering of the input data. Finally, Reinforcement learning dynamically interacts with its environment and it receives positive or negative feedback to improve

its performance.

***Fig: 4.1 Process of machine learning***

#### **4.1.2 Anaconda Navigator**

Anaconda Navigator is a desktop graphical user interface (GUI) included in

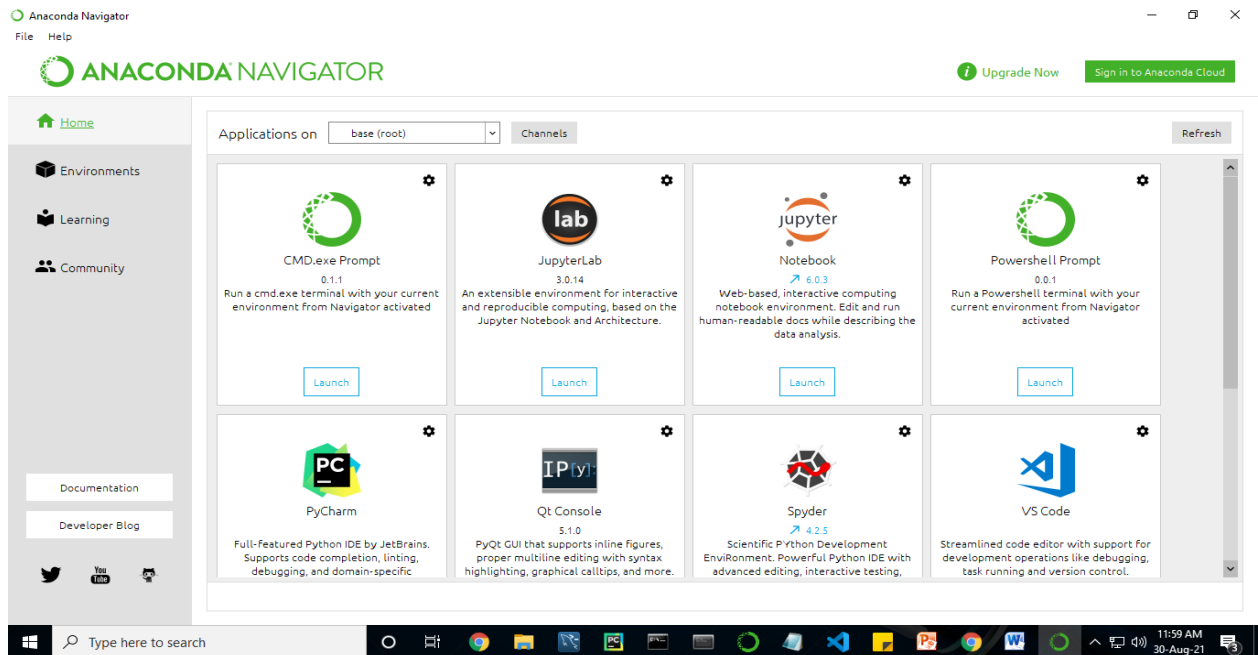


**Process Of Machine Learning**

Anaconda distribution that allows you to launch applications and easily manage conda packages, environments, and channels without using command-line commands. Navigator can search for packages on Anaconda.org or in a local Anaconda Repository.

Anaconda. Now, if you are primarily doing data science work, Anaconda is also a great option. Anaconda is created by Continuum Analytics, and it is a Python distribution that comes preinstalled with lots of useful python libraries for data science.

Anaconda is a distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment.



**Figure 4.2 Anaconda Navigator**

### 4.1.3 JUPYTER NOTEBOOK

This website acts as “meta” documentation for the Jupyter ecosystem. It has a collection of resources to navigate the tools and communities in this ecosystem, and to help you get started.

Project Jupyter is a project and community whose goal is to "develop open-source software, open-standards, and services for interactive computing across dozens of programming languages". It was spun off from IPython in 2014 by Fernando Perez.

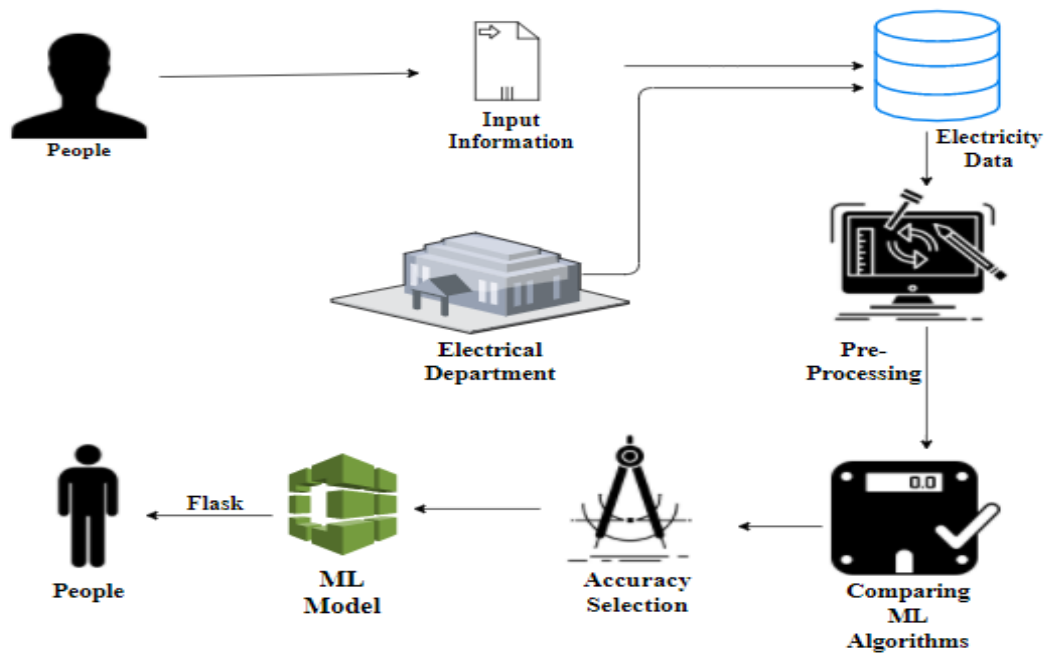
Notebook documents are documents produced by the Jupyter Notebook App, which contain both computer code (e.g. python) and rich text elements (paragraph, equations, figures, links, etc...).

#### **4.1.4 PYTHON**

Python is an interpreted high-level general-purpose programming language. Its design philosophy emphasizes code readability with its use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.

## **4.2 SYSTEM ARCHITECTURE**



**Fig: 4.3 System Architecture**

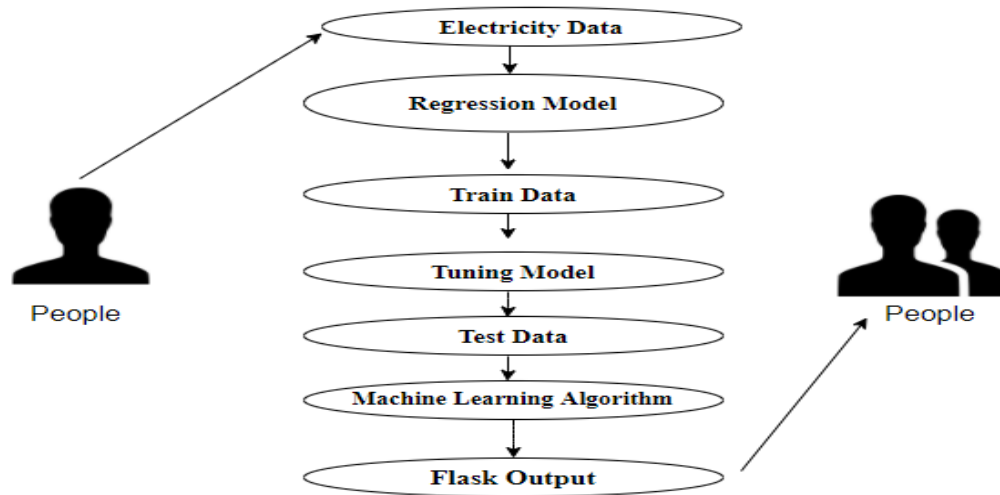
Before beginning the cycle, data will be gathered for those engaged with the venture and composed as a CSV document. The gathered information is then put away in a confined area for some time in the future. This assortment of data is not extensive and contains essential data. This incorporates data before handling, perusing, and separating data for nothing. This strategy for sifting data cannot be given in machine preparation; otherwise, the eventual outcomes will not be valid. Erasing these pointless qualities is the actual data placed into the process. Indeed, even at this stage, the posted data is a CSV document that customary individuals cannot comprehend. Here the data is ready, and the laid-out data is given in the drawings. It is presently straightforward so that the client can discover the information in the informational index. It shows how attractive data is, permitting clients to work with it better. After the information is shown, demonstrating how the introduced information will be addressed, the cycle continuously looks at the calculation. To prepare the information given by the machine, look at the changed calculations and discover the calculation that gives the genuine outcomes. After carrying out every calculation, select the fitting calculation to show the genuine outcomes. The statistics are then led,

and the clients are shown the correct estimation of the power bill for money.

### ***Work flow diagram***

***Fig: 4.4 Workflow diagram***

### ***Use Case Diagram***



***Fig: 4.5 Use case diagram***

Use case diagrams are considered for high level requirement analysis of a system. So when the requirements of a system are analyzed the functionalities are captured in use cases. So, it can say that uses cases are nothing but the system functionalities written in an organized manner.

## **4.3 MODULE DESCRIPTION**

### **4.3.1 Data Pre-Processing**

Validation techniques in machine learning are used to get the error rate of the Machine Learning (ML) model, which can be considered as close to the true error rate of the dataset. If the data volume is large enough to be representative of the population, you may not need the validation techniques. However, in real-world scenarios, to work with samples of data that may not be a true representative of the population of given dataset. To finding the missing value, duplicate value and description of data type whether it is float variable or integer. The sample of data used to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyper parameters.

The evaluation becomes more biased as skill on the validation dataset is incorporated into the model configuration. The validation set is used to evaluate a given model, but this is for frequent evaluation. It as machine learning engineers use

this data to fine-tune the model hyper parameters. Data collection, data analysis, and the process of addressing data content, quality, and structure can add up to a time-consuming to-do list. During the process of data identification, it helps to understand your data and its properties; this knowledge will help you choose which algorithm to use to build your model.

A number of different data cleaning tasks using Python's Pandas library and specifically, it focus on probably the biggest data cleaning task, missing values and it able to more quickly clean data. It wants to spend less time cleaning data, and more time exploring and modeling.

Some of these sources are just simple random mistakes. Other times, there can be a deeper reason why data is missing. It's important to understand these different types of missing data from a statistics point of view. The type of missing data will influence how to deal with filling in the missing values and to detect missing values, and do some basic imputation and detailed statistical approach for dealing with missing data. Before, joint into code, it's important to understand the sources of missing data. Here are some typical reasons why data is missing:

- User forgot to fill in a field.
- Data was lost while transferring manually from a legacy database.
- There was a programming error.
- Users chose not to fill out a field tied to their beliefs about how the results would be used or interpreted.

Variable identification with Uni-variate, Bi-variate and Multi-variate analysis:

- ☐ import libraries for access and functional purpose and read the given dataset
- ☐ General Properties of Analyzing the given dataset
- ☐ Display the given dataset in the form of data frame
- ☐ show columns
- ☐ shape of the data frame
- ☐ To describe the data frame
- ☐ Checking data type and information about dataset

- ☐ Checking for duplicate data
- ☐ Checking Missing values of data frame
- ☐ Checking unique values of data frame
- ☐ Checking count values of data frame
- ☐ Rename and drop the given data frame
- ☐ To specify the type of values
- ☐ To create extra columns

#### 4.3.2 Data Validation/ Cleaning/Preparing Process

Importing the library packages with loading given dataset. To analyzing the variable identification by data shape, data type and evaluating the missing values, duplicate values. A validation dataset is a sample of data held back from training your model that is used to give an estimate of model skill while tuning model's and procedures that you can use to make the best use of validation and test datasets when evaluating your models. Data cleaning / preparing by rename the given dataset and drop the column etc. to analyze the uni-variate, bi-variate and multi-variate process. The steps and techniques for data cleaning will vary from dataset to dataset. The primary goal of data cleaning is to detect and remove errors and anomalies to increase the value of data in analytics and decision making.

|                    | ATTRIBUTES                  |
|--------------------|-----------------------------|
| num_rooms          | No. of rooms                |
| num_people         | Number of people            |
| housearea          | Area of house               |
| is_ac              | Presence of AC              |
| is_tv              | Presence of television      |
| is_flat            | Is residence flat or not    |
| ave_monthly_income | Monthly income              |
| num_children       | No of children in household |
| is_urban           | Residence in urban or rural |
| amount_paid        | The amount paid             |

**Table: 4.1 Attributes**

```
In [1]: import pandas as pd
import numpy as np

In [2]: import warnings
warnings.filterwarnings("ignore")

In [3]: data = pd.read_csv("Household energy bill data.csv")

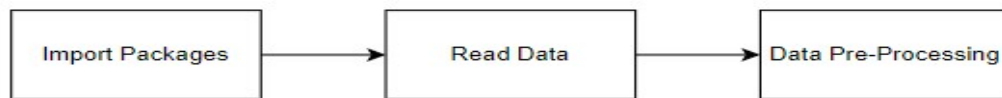
Before drop the given dataset:

In [4]: data.head()
Out[4]:
```

|   | num_rooms | num_people | housearea | is_ac | is_tv | is_flat | ave_monthly_income | num_children | is_urban | amount_paid |
|---|-----------|------------|-----------|-------|-------|---------|--------------------|--------------|----------|-------------|
| 0 | 3         | 3          | 742.57    | 1     | 1     | 1       | 9675.93            | 2            | 0        | 560.481447  |
| 1 | 1         | 5          | 952.99    | 0     | 1     | 0       | 35064.79           | 1            | 1        | 633.283679  |
| 2 | 3         | 1          | 761.44    | 1     | 1     | 1       | 22292.44           | 0            | 0        | 511.879157  |
| 3 | 0         | 5          | 861.32    | 1     | 1     | 0       | 12139.08           | 0            | 0        | 332.992035  |
| 4 | 1         | 8          | 731.61    | 0     | 1     | 0       | 17230.10           | 2            | 1        | 658.285625  |

**Fig: 4.6 Module 1 Data validation**

**Module diagram:**



**Fig: 4.7 Module 1 Diagram**

**Given Input Expected Output**

input : data

output : removing noisy data

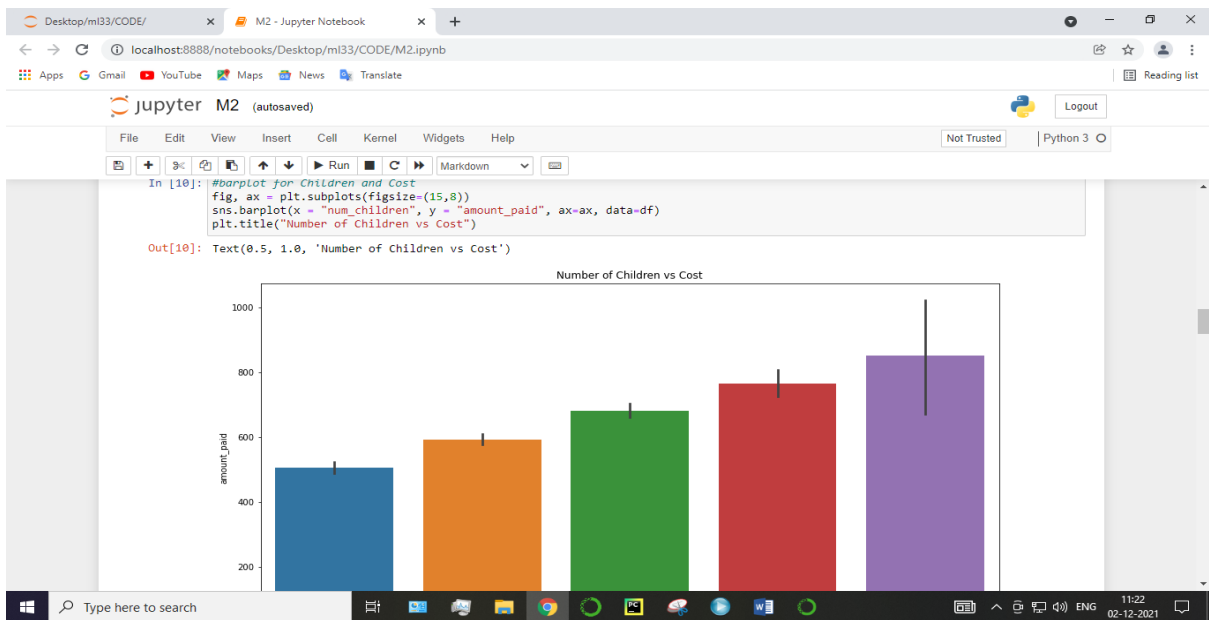
### 4.3.3 Exploration Data Analysis Of Visualization

Data visualization is an important skill in applied statistics and machine learning. Statistics does indeed focus on quantitative descriptions and estimations of data. Data visualization provides an important suite of tools for gaining a qualitative understanding. This can be helpful when exploring and getting to know a dataset and can help with identifying patterns, corrupt data, outliers, and much more. With a little domain knowledge, data visualizations can be used to express and demonstrate key

relationships in plots and charts that are more visceral and stakeholders than measures of association or significance. Data visualization and exploratory data analysis are whole fields themselves and it will recommend a deeper dive into some the books mentioned at the end.

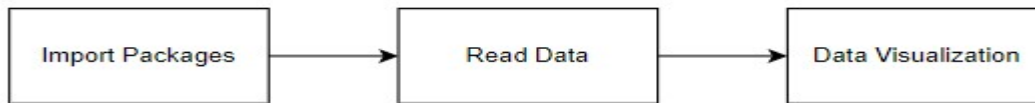
Sometimes data does not make sense until it can look at in a visual form, such as with charts and plots. Being able to quickly visualize of data samples and others is an important skill both in applied statistics and in applied machine learning. It will discover the many types of plots that you will need to know when visualizing data in Python and how to use them to better understand your own data.

- How to chart time series data with line plots and categorical quantities with bar charts.
- How to summarize data distributions with histograms and box plots.



**Fig: 4.8 Module 2 Visualization**

### **Module diagram:**



**Fig: 4.9 Module 2 Diagram**

### **Given Input Expected Output**

input : data

output : visualized data

Pre-processing refers to the transformations applied to our data before feeding it to the algorithm. Data Preprocessing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis. To achieving better results from the applied model in Machine Learning method of the data has to be in a proper manner. Some specified Machine Learning model needs information in a specified format, for example, Random Forest algorithm does not support null values. Therefore, to execute random forest algorithm null values have to be managed from the original raw data set. And another aspect is that data set should be formatted in such a way that more than one Machine Learning and Deep Learning algorithms are executed in given dataset.

### **4.3.4 Comparing Algorithm With Prediction In The Form Of Best Accuracy Result**

It is important to compare the performance of multiple different machine learning algorithms consistently and it will discover to create a test harness to compare multiple different machine learning algorithms in Python with scikit-learn. It can use this test harness as a template on your own machine learning problems and add more and different algorithms to compare. Each model will have different performance characteristics. Using resampling methods like cross validation, you can get an estimate for how accurate each model may be on unseen data. It needs to be able to

use these estimates to choose one or two best models from the suite of models that you have created. When have a new dataset, it is a good idea to visualize the data using different techniques in order to look at the data from different perspectives. The same idea applies to model selection. You should use a number of different ways of looking at the estimated accuracy of your machine learning algorithms in order to choose the one or two to finalize. A way to do this is to use different visualization methods to show the average accuracy, variance and other properties of the distribution of model accuracies.

In the next section you will discover exactly how you can do that in Python with scikit-learn. The key to a fair comparison of machine learning algorithms is ensuring that each algorithm is evaluated in the same way on the same data and it can achieve this by forcing each algorithm to be evaluated on a consistent test harness.

In the example below 6 different algorithms are compared:

- ☐ Support Vector Regression
- ☐ Decision Tree Regression
- ☐ Random Forest Regression
- ☐ Linear Regression
- ☐ Lasso Regression
- ☐ Ridge Regression

The K-fold cross validation procedure is used to evaluate each algorithm, importantly configured with the same random seed to ensure that the same splits to the training data are performed and that each algorithm is evaluated in precisely the same way. Before that comparing algorithm, Building a Machine Learning Model using install Scikit-Learn libraries. In this library package have to done preprocessing, linear model with logistic regression method, cross validating by KFold method, ensemble with random forest method and tree with decision tree classifier. Additionally, splitting the train set and test set. To predicting the result by comparing accuracy.

#### 4.3.5 Regression:

Regression is a statistical method used in finance, investing, and other disciplines that attempts to determine the strength and character of the relationship between one dependent variable (usually denoted by  $Y$ ) and a series of other variables (known as independent variables).

Regression helps investment and financial managers to value assets and understand the relationships between variables, such as commodity prices and the stocks of businesses dealing in those commodities.

The two basic types of regression are simple linear regression and multiple linear regression, although there are non-linear regression methods for more complicated data and analysis. Simple linear regression uses one independent variable to explain or predict the outcome of the dependent variable  $Y$ , while multiple linear regression uses two or more independent variables to predict the outcome.

Regression can help finance and investment professionals as well as professionals in other businesses. Regression can also help predict sales for a company based on weather, previous sales, GDP growth, or other types of conditions. The capital asset pricing model (CAPM) is an often-used regression model in finance for pricing assets and discovering costs of capital.

The general form of each type of regression is:

- **Simple linear regression:**  $Y = a + bX + u$
- **Multiple linear regression:**  $Y = a + b_1X_1 + b_2X_2 + b_3X_3 + \dots + b_tX_t + u$

**where**

- $Y$  = the variable that you are trying to predict (dependent variable).
- $X$  = the variable that you are using to predict  $Y$  (independent variable).
- $a$  = the intercept.
- $b$  = the slope.

- $u$  = the regression residual.

Regression is often used to determine how many specific factors such as the price of a commodity, interest rates, particular industries, or sectors influence the price movement of an asset. The aforementioned CAPM is based on regression, and it is utilized to project the expected returns for stocks and to generate costs of capital. A stock's returns are regressed against the returns of a broader index, such as the S&P 500, to generate a beta for the particular stock.

Beta is the stock's risk in relation to the market or index and is reflected as the slope in the CAPM model. The return for the stock in question would be the dependent variable  $Y$ , while the independent variable  $X$  would be the market risk premium.

Additional variables such as the market capitalization of a stock, valuation ratios, and recent returns can be added to the CAPM model to get better estimates for returns. These additional factors are known as the Fama-French factors, named after the professors who developed the multiple linear regression model to better explain asset returns.

Multiple linear regression (MLR), also known simply as multiple regression, is a statistical technique that uses several explanatory variables to predict the outcome of a response variable. The goal of multiple linear regression is to model the linear relationship between the explanatory (independent) variables and response (dependent) variables. In essence, multiple regression is the extension of ordinary least-squares (OLS) regression because it involves more than one explanatory variable.

Simple linear regression is a function that allows an analyst or statistician to make predictions about one variable based on the information that is known about another variable. Linear regression can only be used when one has two continuous variables—an independent variable and a dependent variable. The independent variable is the parameter that is used to calculate the dependent variable or outcome. A multiple regression model extends to several explanatory variables.

## 4.4 ALGORITHM AND TECHNIQUES

### ***Algorithm Explanation***

In machine learning and statistics, classification is a supervised learning approach in which the computer program learns from the data input given to it and then uses this learning to classify new observation. This data set may simply be bi-class (like identifying whether the person is male or female or that the mail is spam or non-spam) or it may be multi-class too. Some examples of classification problems are: speech recognition, handwriting recognition, bio metric identification, document classification etc. In Supervised Learning, algorithms learn from labeled data. After understanding the data, the algorithm determines which label should be given to new data based on pattern and associating the patterns to the unlabeled new data.

### ***Used Python Packages:***

#### ***sklearn:***

- In python, sklearn is a machine learning package which include a lot of ML algorithms.
- Here, we are using some of its modules like train\_test\_split, DecisionTreeClassifier or Logistic Regression and accuracy\_score.

#### ***NumPy:***

- It is a numeric python module which provides fast maths functions for calculations.
- It is used to read data in numpy arrays and for manipulation purpose.

#### ***Pandas:***

- Used to read and write different files.
- Data manipulation can be done easily with data frames.

#### ***Matplotlib:***

- Data visualization is a useful way to help with identify the patterns from given dataset.

- Data manipulation can be done easily with data frames.

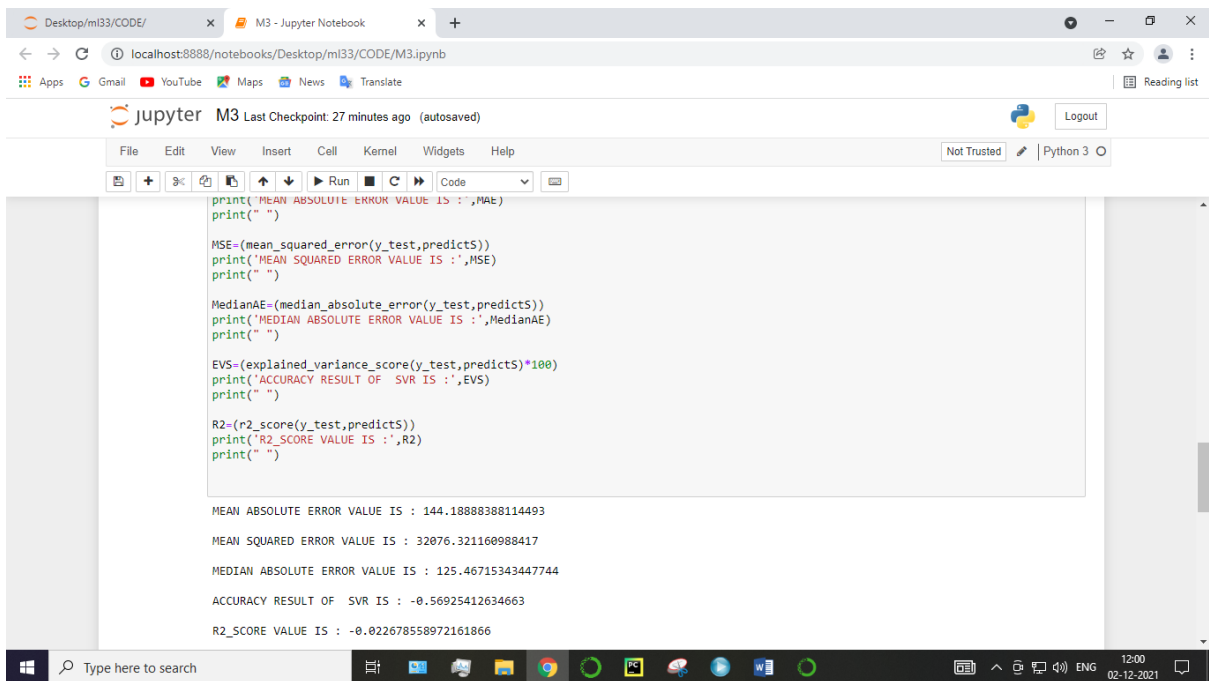
#### 4.4.1 Support Vector Regression:

Support Vector Machines (SVM) are popularly and widely used for classification and regression problems in machine learning.

The regression problem is a generalization of the classification problem, in which the model returns a continuous-valued output, as opposed to an output from a finite set. In other words, a regression model estimates a continuous-valued multivariate function.

SVMs solve binary classification problems by formulating them as convex optimization problems. The optimization problem entails finding the maximum margin separating the hyperplane, while correctly classifying as many training points as possible. SVMs represent this optimal hyperplane with support vectors. The sparse solution and good generalization of the SVM lend themselves to adaptation to regression problems. SVM generalization to SVR is accomplished by introducing an  $\epsilon$ -insensitive region around the function, called the  $\epsilon$ -tube. This tube reformulates the optimization problem to find the tube that best approximates the continuous-valued function, while balancing model complexity and prediction error. More specifically, SVR is formulated as an optimization problem by first defining a convex  $\epsilon$ -insensitive loss function to be minimized and finding the flattest tube that contains most of the training instances. Hence, a multi-objective function is constructed from the loss function and the geometrical properties of the tube. Then, the convex optimization, which has a unique solution, is solved, using appropriate numerical optimization algorithms. The hyperplane is represented in terms of support vectors, which are training samples that lie outside the boundary of the tube. As in SVM, the support vectors in SVR are the most influential instances that affect the shape of the tube, and the training and test data are assumed to be independent and identically distributed (iid), drawn from the same fixed but unknown probability distribution function in a

supervised-learning context.



```
print('MEAN ABSOLUTE ERROR VALUE IS :',MAE)
print(" ")
MSE=(mean_squared_error(y_test,predicts))
print('MEAN SQUARED ERROR VALUE IS :',MSE)
print(" ")
MedianAE=(median_absolute_error(y_test,predicts))
print('MEDIAN ABSOLUTE ERROR VALUE IS :',MedianAE)
print(" ")
EVS=(explained_variance_score(y_test,predicts)*100)
print('ACCURACY RESULT OF SVR IS :',EVS)
print(" ")
R2=(r2_score(y_test,predicts))
print('R2_SCORE VALUE IS :',R2)
print(" ")
```

MEAN ABSOLUTE ERROR VALUE IS : 144.18888388114493

MEAN SQUARED ERROR VALUE IS : 32076.321160988417

MEDIAN ABSOLUTE ERROR VALUE IS : 125.46715343447744

ACCURACY RESULT OF SVR IS : -0.56925412634663

R2\_SCORE VALUE IS : -0.022678558972161866

**Fig: 4.10 Module 3 Support Vector Regression**

#### 4.4.2 Decision Tree Regression:

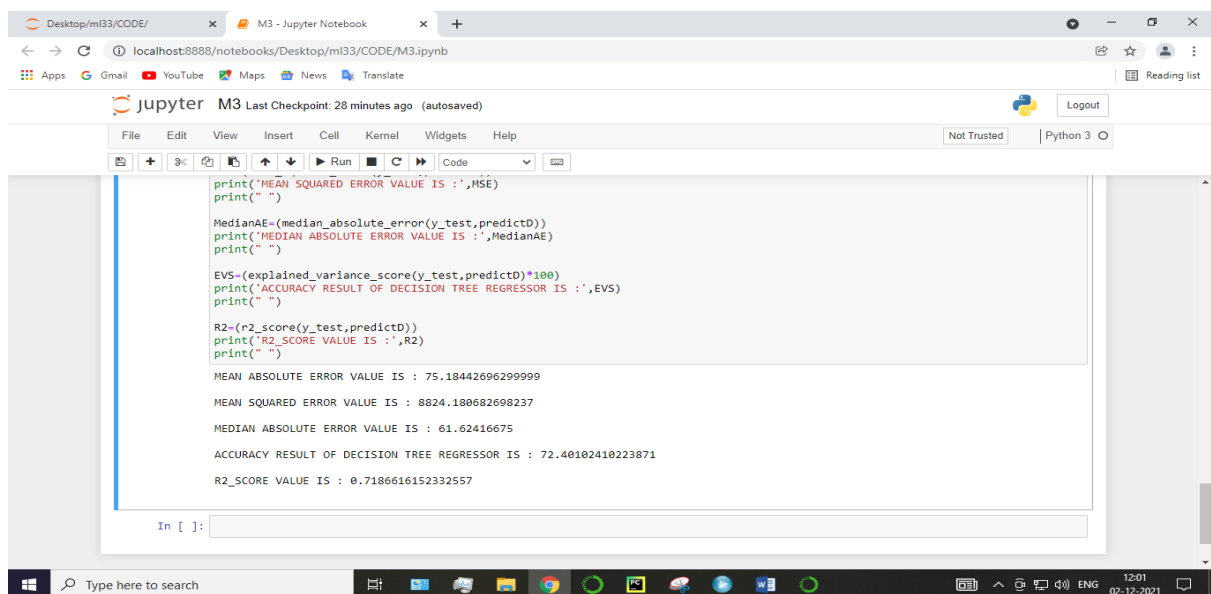
Decision Tree - Regression. Decision tree builds regression or classification models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes.

Decision tree builds regression or classification models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed.

The final result is a tree with decision nodes and leaf nodes. A decision node (e.g., Outlook) has two or more branches (e.g., Sunny, Overcast and Rainy), each representing values for the attribute tested. Leaf node (e.g., Hours Played) represents a decision on the numerical target. The topmost decision node in a tree which

corresponds to the best predictor called root node. Decision trees can handle both categorical and numerical data.

Decision Tree is one of the most commonly used, practical approaches for supervised learning. It can be used to solve both Regression and Classification tasks with the latter being put more into practical application. It is a tree-structured classifier with three types of nodes. Decision trees where the target variable can take continuous values (typically real numbers) are called regression trees. Decision trees are among the most popular machine learning algorithms given their intelligibility and simplicity.

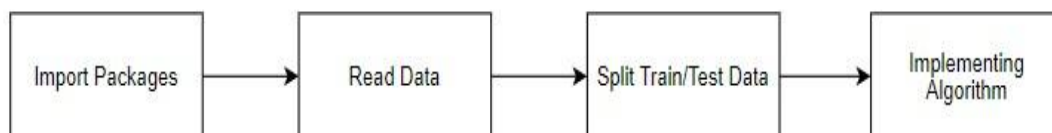


```
print('MEAN SQUARED ERROR VALUE IS :',MSE)
print(" ")
MedianAE=(median_absolute_error(y_test,predictD))
print('MEDIAN ABSOLUTE ERROR VALUE IS :',MedianAE)
print(" ")
EVS=(explained_variance_score(y_test,predictD)*100)
print('ACCURACY RESULT OF DECISION TREE REGRESSOR IS :',EVS)
print(" ")
R2=(r2_score(y_test,predictD))
print('R2_SCORE VALUE IS :',R2)
print(" ")

MEAN ABSOLUTE ERROR VALUE IS : 75.18442696299999
MEAN SQUARED ERROR VALUE IS : 8824.180682698237
MEDIAN ABSOLUTE ERROR VALUE IS : 61.62416675
ACCURACY RESULT OF DECISION TREE REGRESSOR IS : 72.40102410223871
R2_SCORE VALUE IS : 0.7186616152332557
```

**Fig: 4.11 Module 3 Decision Tree Regressor**

## Module Diagram



**Fig: 4.12 Module 3 Diagram**

### ***Given Input Expected Output***

Input : Data

Output : Getting Accuracy

#### **4.4.3 Random Forest Regression:**

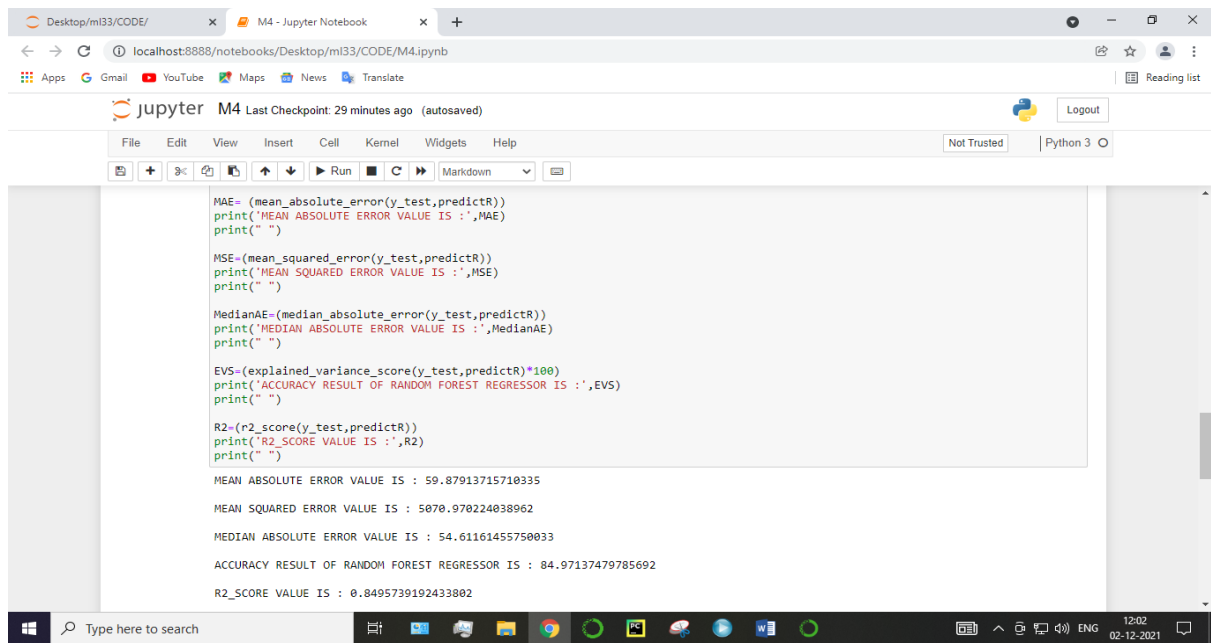
Random Forest Regression is a supervised learning algorithm that uses ensemble learning method for regression.

Ensemble learning method is a technique that combines predictions from multiple machine learning algorithms to make a more accurate prediction than a single model. We have a basic understanding of how the Random Forest Regression model works, we can assess its performance on a real-world dataset. Similar to my previous posts, I will be using data on House Sales in King County, USA.

After importing the libraries, importing the dataset, addressing null values, and dropping any necessary columns, we are ready to create our Random Forest Regression model. From the sklearn package containing ensemble learning, we import the class Random Forest Regressor, create an instance of it, and assign it to a variable. The parameter `n_estimators` creates  $n$  number of trees in your random forest, where  $n$  is the number you pass in. We passed in 10.

The `.fit()` function allows us to train the model, adjusting weights according to the data values in order to achieve better accuracy. After training, our model is ready to make predictions, which is called by the `.predict()` method.

A random forest is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is controlled with the `max_samples` parameter if `Bootstrap = True` (default), otherwise the whole dataset is used to build each tree.



```
MAE=(mean_absolute_error(y_test,predictR))
print('MEAN ABSOLUTE ERROR VALUE IS :',MAE)
print(" ")

MSE=(mean_squared_error(y_test,predictR))
print('MEAN SQUARED ERROR VALUE IS :',MSE)
print(" ")

MedianAE=(median_absolute_error(y_test,predictR))
print('MEDIAN ABSOLUTE ERROR VALUE IS :',MedianAE)
print(" ")

EVS=(explained_variance_score(y_test,predictR)*100)
print('ACCURACY RESULT OF RANDOM FOREST REGRESSOR IS :',EVS)
print(" ")

R2=(r2_score(y_test,predictR))
print('R2_SCORE VALUE IS :',R2)
print(" ")

MEAN ABSOLUTE ERROR VALUE IS : 59.87913715710335

MEAN SQUARED ERROR VALUE IS : 5070.970224038962

MEDIAN ABSOLUTE ERROR VALUE IS : 54.61161455750033

ACCURACY RESULT OF RANDOM FOREST REGRESSOR IS : 84.97137479785692

R2_SCORE VALUE IS : 0.8495739192433802
```

**Fig: 4.13 Module 4 Random Forest Regressor**

#### 4.4.4 Linear Regression:

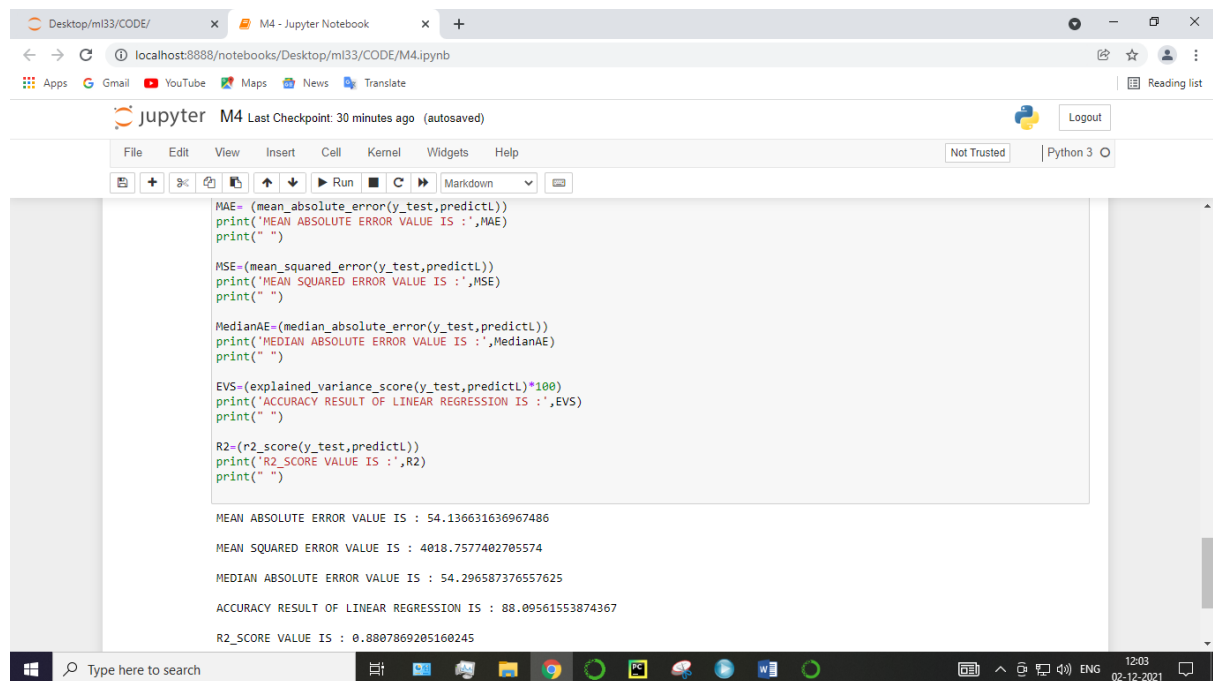
Linear Regression is a machine learning algorithm based on supervised learning. Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x). So, this regression technique finds out a linear relationship between x (input) and y(output).

It performs a regression task. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting. Different regression models differ based on the kind of relationship between dependent and independent variables, they are considering and the number of independent variables being used.

Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x). So, this regression technique finds out a linear relationship between x (input) and y(output). Hence, the name is Linear Regression. In the figure above, X (input) is the work experience and Y (output) is the work experience and Y (output) is the salary of a person. The regression line is the

best fit line for our model. Linear regression is one of the easiest and most popular Machine Learning algorithms. It is a statistical method that is used for predictive analysis. Linear regression makes predictions for continuous/real or numeric variables such as sales, salary, age, product price, etc.

Linear regression algorithm shows a linear relationship between a dependent (y) and one or more independent (x) variables, hence called as linear regression. Since linear regression shows the linear relationship, which means it finds how the value of the dependent variable is changing according to the value of the independent variable. The linear regression model provides a sloped straight line representing the relationship between the variables.



The screenshot displays a Jupyter Notebook window titled 'M4 - Jupyter Notebook' running on a local host. The notebook contains a single code cell with the following Python code:

```
MAE=(mean_absolute_error(y_test,predictL))
print('MEAN ABSOLUTE ERROR VALUE IS :',MAE)
print(" ")

MSE=(mean_squared_error(y_test,predictL))
print('MEAN SQUARED ERROR VALUE IS :',MSE)
print(" ")

MedianAE=(median_absolute_error(y_test,predictL))
print('MEDIAN ABSOLUTE ERROR VALUE IS :',MedianAE)
print(" ")

EVS=(explained_variance_score(y_test,predictL)*100)
print('ACCURACY RESULT OF LINEAR REGRESSION IS :',EVS)
print(" ")

R2=(r2_score(y_test,predictL))
print('R2_SCORE VALUE IS :',R2)
print(" ")
```

The output of the code is displayed below the code cell:

```
MEAN ABSOLUTE ERROR VALUE IS : 54.136631636967486

MEAN SQUARED ERROR VALUE IS : 4018.7577402705574

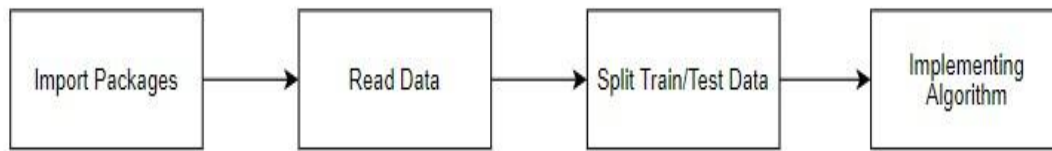
MEDIAN ABSOLUTE ERROR VALUE IS : 54.296587376557625

ACCURACY RESULT OF LINEAR REGRESSION IS : 88.09561553874367

R2_SCORE VALUE IS : 0.8807869205160245
```

**Fig: 4.14 Module 4 Linear Regression**

### Module diagram:



**Fig 4.15 Module 4 Diagram**

### ***Given Input Expected Output***

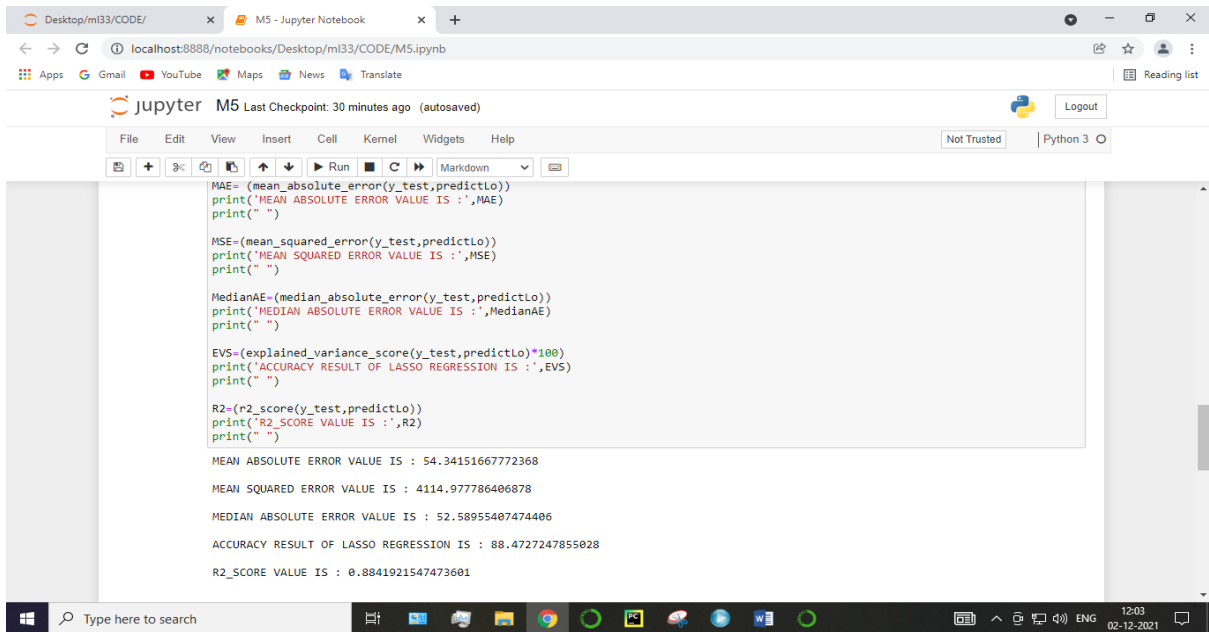
Input : Data

Output : Getting Accuracy

### **4.4.5 Lasso Regression:**

Lasso regression is a regularization technique. It is used over regression methods for a more accurate prediction. This model uses shrinkage. Shrinkage is where data values are shrunk towards a central point as the mean. The lasso procedure encourages simple, sparse models (i.e. models with fewer parameters). The word “LASSO” stands for Least absolute Shrinkage and Selection operator. It is a statistical formula for the regularisation of data models and feature selection. Lasso regression is a regularization technique. It is used over regression methods for a more accurate prediction. This model uses shrinkage. Shrinkage is where data values are shrunk towards a central point as the mean. The lasso procedure encourages simple, sparse models (i.e. models with fewer parameters). This particular type of regression is well-suited for models showing high levels of multicollinearity or when you want to automate certain parts of model selection, like variable selection/parameter elimination. Lasso Regression uses L1 regularization technique (will be discussed later in this article). It is used when we have more number of features because it automatically performs feature selection. The L1 regularization performed by Lasso, causes the regression coefficient of the less contributing variable to shrink to zero or near zero. Lasso regression uses shrinkage, where the data values are shrunk

towards a central point such as the mean value. The Lasso penalty shrinks or reduces the coefficient value towards zero. The less contributing variable is therefore allowed to have a zero or near-zero coefficient.



The screenshot shows a Jupyter Notebook window titled 'M5 - Jupyter Notebook' with the URL 'localhost:8888/notebooks/Desktop/m133/CODE/M5.ipynb'. The notebook interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, running cells, and markdown editing. The code cell contains the following Python code:

```
MAE=(mean_absolute_error(y_test,predictIo))
print('MEAN ABSOLUTE ERROR VALUE IS :',MAE)
print(" ")

MSE=(mean_squared_error(y_test,predictIo))
print('MEAN SQUARED ERROR VALUE IS :',MSE)
print(" ")

MedianAE=(median_absolute_error(y_test,predictIo))
print('MEDIAN ABSOLUTE ERROR VALUE IS :',MedianAE)
print(" ")

EV5=(explained_variance_score(y_test,predictIo)*100)
print('ACCURACY RESULT OF LASSO REGRESSION IS :',EV5)
print(" ")

R2=(r2_score(y_test,predictIo))
print('R2_SCORE VALUE IS :',R2)
print(" ")
```

The output of the code is displayed below the code cell:

```
MEAN ABSOLUTE ERROR VALUE IS : 54.34151667772368
MEAN SQUARED ERROR VALUE IS : 4114.977786406878
MEDIAN ABSOLUTE ERROR VALUE IS : 52.58955407474406
ACCURACY RESULT OF LASSO REGRESSION IS : 88.4727247855028
R2_SCORE VALUE IS : 0.8841921547473601
```

**Fig: 4.16 Module 5 Lasso regression**

#### 4.4.6 Ridge Regression:

Ridge regression is a model tuning method that is used to analyse any data that suffers from multicollinearity. This method performs L2 regularization. When the issue of multicollinearity occurs, least-squares are unbiased, and variances are large, this results in predicted values to be far away from the actual values.

Ridge Regression Works By Increasing The Bias To Improve Variance. This Works By Changing The Slope Of The Line. The Model Performance Might Be Little Poor On The Training Set But It Will Perform Consistently Well On Both The Training And Testing Dataset. Slope Has Been Reduced With Ridge Regression Penalty(A) And Therefore The Model Becomes Less Sensitive To Change In The Independent Variable

The screenshot shows a Jupyter Notebook interface with a code cell containing the following Python code:

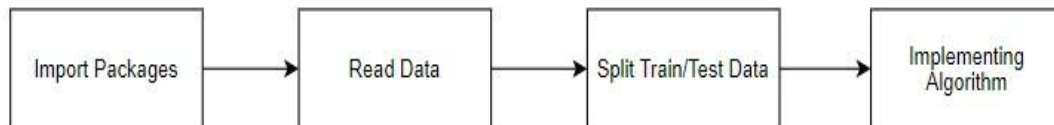
```
print("\n")
MSE=(mean_squared_error(y_test,predictr1))
print('MEAN SQUARED ERROR VALUE IS :',MSE)
print("\n")
MedianAE=(median_absolute_error(y_test,predictr1))
print('MEDIAN ABSOLUTE ERROR VALUE IS :',MedianAE)
print("\n")
EV5=(explained_variance_score(y_test,predictr1)*100)
print('ACCURACY RESULT OF RIDGE REGRESSION IS :',EV5)
print("\n")
R2=(r2_score(y_test,predictr1))
print('R2_SCORE VALUE IS :',R2)
print("\n")
```

The output of the code is displayed below the cell:

```
MEAN ABSOLUTE ERROR VALUE IS : 54.33456524402068
MEAN SQUARED ERROR VALUE IS : 4101.01695177962
MEDIAN ABSOLUTE ERROR VALUE IS : 52.16278337959034
ACCURACY RESULT OF RIDGE REGRESSION IS : 88.52340290913202
R2_SCORE VALUE IS : 0.8845850546024825
```

**Fig: 4.17 Module 5 Ridge Regression**

**Module diagram:**



**Fig: 4.18 Module 5 Diagram**

**Given Input Expected Output**

input : data

output : getting accuracy

## 4.5 DEPLOYMENT

### 4.5.1 Flask (Web Framework) :

Framework Flask is a web framework from Python language. Flask provides a library and a collection of codes that can be used to build websites, without the need to do everything from scratch. But Framework flask still doesn't use the Model View Controller (MVC) method.

Flask-RESTful is an extension for Flask that provides additional support for building REST APIs. You will never be disappointed with the time it takes to develop an API. Flask-Restful is a lightweight abstraction that works with the existing ORM/libraries. Flask-RESTful encourages best practices with minimal setup.

Flask Restful is an extension for Flask that adds support for building REST APIs in Python using Flask as the back-end. It encourages best practices and is very easy to set up. Flask restful is very easy to pick up if you're already familiar with flask.

Flask is a web framework for Python, meaning that it provides functionality for building web applications, including managing HTTP requests and rendering templates and also we can add to this application to create our API.

#### ***Advantages of Flask:***

- Higher compatibility with latest technologies.
- Technical experimentation.
- Easier to use for simple cases.
- Codebase size is relatively smaller.
- High scalability for simple applications.
- Easy to build a quick prototype.
- Routing URL is easy.
- Easy to develop and maintain applications.

### 4.5.2 HTML

HTML stands for Hyper Text Markup Language. It is used to design web pages using a markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages. A markup language is used to define the text document within tag which defines the structure of web pages. This language is used to annotate (make notes for the computer) text so that a machine can understand it and manipulate text accordingly. Most markup languages (e.g. HTML) are human-readable. The language uses tags to define what manipulation has to be done on the text.

### **4.5.3 CSS**

CSS stands for Cascading Style Sheets. It is the language for describing the presentation of Web pages, including colours, layout, and fonts, thus making our web pages presentable to the users. CSS is designed to make style sheets for the web. It is independent of HTML and can be used with any XML-based markup language. Now let's try to break the acronym:

- Cascading: Falling of Styles
- Style: Adding designs/Styling our HTML tags
- Sheets: Writing our style in different documents

## **CHAPTER – 5**

## RESULTS AND DISCUSSION

### 5.1 RESULT

|                           |                   |
|---------------------------|-------------------|
| Support Vector Regression | -0.569254126      |
| Decision Tree Regression  | 72.40102410       |
| Random Forest Regression  | 84.9713748        |
| Linear Regression method  | 88.09561553874367 |
| Lasso Regression          | 88.4727247855028  |
| Ridge Regression          | 88.52340290913202 |

**Table: 5.1 Accuracy Results**

The above table defines the accurateness at which each algorithm can predict electricity bills with sample data. The most accurate algorithm is discovered using these percentage values, and it is used during prediction. Then the algorithm is deployed in the flask and implemented in the local host for prediction. After entering the input data, the most accurate algorithm is implemented to find precise results. In this case, the most accurate algorithm is Ridge Regression with 88.52340290913202% accuracy. The input data is implemented with the Ridge Regression algorithm, obtaining accurate results.

The screenshot shows a web application titled "ELECTRICITY BILL PREDICTION". The interface features a dark background with a blue lightning bolt graphic. On the left side, there are input fields for "NUMBER OF ROOMS" (value: 2), "NUMBER OF PEOPLE" (value: 3), "HOUSE AREA" (value: 654), "IS AC" (value: 1), and "IS TV" (value: 1). On the right side, there are input fields for "IS FLAT" (value: 0), "MONTHLY INCOME" (value: 50000), "NUMBER OF CHILDREN" (value: 0), and "IS URBAN" (value: 1). A green "Predict" button is located at the bottom center of the form.

**Fig: 5.1 Output screenshot**

NUMBER OF ROOMS  
NUMBER OF ROOMS

NUMBER OF PEOPLE  
NUMBER OF PEOPLE

HOUSE AREA  
HOUSE AREA

IS AC  
IS AC

IS TV  
IS TV

IS FLAT  
IS FLAT

AVERAGE MONTHLY INCOME  
AVERAGE MONTHLY INCOME

NUMBER OF CHILDREN  
NUMBER OF CHILDREN

IS URBAN  
IS URBAN

Predict

693.07 AMOUNT OF ELECTRICITY BILL  
GENERATED

***Fig: 5.2 Output screenshot***

## 5.2 DISCUSSION

After comparing all the six algorithm, the Ridge Regression algorithm got the most accurate value so the algorithm is deployed using flask and implemented in the local host for prediction. In the above output with the help of local host a new tab opens to input the values of Number of rooms, Number of people, House area, Is AC, Is TV, Is Flat, Average monthly income, Number of children, Is Urban. After giving all the values press Predict button to predict the Electricity bill, then in the bottom of the window we will get the accurate amount of electricity bill will be generated.

## **CHAPTER – 6**

### **CONCLUSION AND FUTURE WORK**

#### **28. CONCLUSION**

The analytical process started from data cleaning and processing, missing value, exploratory analysis and finally model building and evaluation. The best accuracy on public test set is higher accuracy score is will be find out. This application can help to find the Electricity Bill.

#### **29. FUTURE WORK**

- ☐ Electricity Bill prediction to connect with AI model.
- ☐ To automate this process by show the prediction result in web application or desktop application at cloud.
- ☐ To optimize the work to implement in Artificial Intelligence environment.

## REFERENCES

1. S. K. Aggarwal, L. M. Saini, and A. Kumar, "Electricity price forecasting in deregulated markets: A review and evaluation," *International Journal of Electrical Power & Energy Systems*, vol. 31, no. 1, pp. 13–22, Jan. 2009, doi: 10.1016/j.ijepes.2008.09.003.
2. G. C. McDonald, "Ridge regression," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 1, no. 1, pp. 93–100, Jul. 2009, doi: 10.1002/wics.14.
3. A. Lucas, K. Pegios, E. Kotsakis, and D. Clarke, "Price Forecasting for the Balancing Energy Market Using Machine-Learning Regression," *Energies*, vol. 13, no. 20, p. 5420, Oct. 2020, doi: 10.3390/en13205420.
4. E. Viviani, L. Di Persio, and M. Ehrhardt, "Energy Markets Forecasting. From Inferential Statistics to Machine Learning: The German Case," *Energies*, vol. 14, no. 2, p. 364, Jan. 2021, doi: 10.3390/en14020364.
5. S. Fan, J. R. Liao, K. Kaneko, and L. Chen, "An Integrated Machine Learning Model for Day-Ahead Electricity Price Forecasting," *IEEE Xplore*, Oct. 01, 2006. <https://ieeexplore.ieee.org/document/4075985> (accessed Feb. 10, 2022).
6. L. Hu, G. Taylor, H.-B. Wan, and M. Irving, "A review of short-term electricity price forecasting techniques in deregulated electricity markets," *IEEE Xplore*, Sep. 01, 2009. <https://ieeexplore.ieee.org/document/5429485> (accessed Feb. 10, 2022).
7. A. Jain, A. Tuli, and M. Kakkar, "A Review For Electricity Price Forecasting Techniques In Electricity Markets," *International Journal of Engineering Research & Technology*, vol. 2, no. 4, Apr. 2013, Accessed: Feb. 10, 2022. [Online]. Available: <https://www.ijert.org/a-review-for-electricity-price-forecasting-techniques-in-electricity-markets>.
8. Y. Chen, T. Koch, K. G. Lim, X. Xu, and N. Zakiyeva, "A review study of functional autoregressive models with application to energy forecasting," *WIREs Computational Statistics*, vol. 13, no. 3, Jul. 2020, doi: 10.1002/wics.1525.
9. D. M. Titterton and B. Cheng, "[Neural Networks: A Review from Statistical Perspective]: Rejoinder," *Project Euclid*, Feb. 1994, [Online]. Available:

<https://projecteuclid.org/journals/statistical-science/volume-9/issue-1/Neural-Networks-A-Review-from-Statistical-Perspective-Rejoinder/10.1214/ss/1177010646.full?tab=ArticleLink>

10. S. Chen, E. Grant, T. T. Wu, and F. D. Bowman, "Some recent statistical learning methods for longitudinal high-dimensional data," *WIREs Computational Statistics*, vol. 6, no. 1, pp. 10–18, Dec. 2013, doi: 10.1002/wics.1282.
11. Z. Zhang and M. Wu, "Predicting Real-Time Locational Marginal Prices: A GAN-Based Approach," *IEEE Transactions on Power Systems*, pp. 1–1, Aug. 2021, doi: 10.1109/TPWRS.2021.3106263.
12. X. Yan and X. G. Su, "Linear Regression Analysis," *World Scientific*, Jun. 2009, doi: 10.1142
13. Suma, V., and Shavige Malleshwara Hills. "Data Mining based Prediction of Demand in Indian Market for Refurbished Electronics." *Journal of Soft Computing Paradigm (JSCP)* 2, no. 02 (2020): 101-110.
14. Anand, C. "Comparison of Stock Price Prediction Models using Pre-trained Neural Networks." *Journal of Ubiquitous Computing and Communication Technologies (UCCT)* 3, no. 02 (2021): 122-134.
15. Sathishkumar, S., R. Devi Priya, and K. Karthika. "Survey on Data Mining and Predictive Analytics Techniques." In *Inventive Communication and Computational Technologies*, pp. 971-981. Springer, Singapore, 2020.